



Universidad del Istmo de Guatemala  
Facultad de Ingenieria  
Ing. en Sistemas  
Informatica II  
Prof. Ernesto Rodriguez - erodriguez@unis.edu.gt

---

## Laboratorio #9

Fecha de entrega: 11 de Abril, 2019 - 11:59pm

---

*Instrucciones: Resolver cada uno de los ejercicios siguiendo sus respectivas instrucciones. El trabajo debe ser entregado a traves de Github, en su repositorio del curso, colocado en una carpeta llamada "Laboratorio #9". Al menos que la pregunta indique diferente, todas las respuestas a preguntas escritas deben presentarse en un documento formato pdf, el cual haya sido generado mediante Latex. Este laboratorio debe ser elaborado en parejas.*

### Tarea #1 (20%)

Declare una *clase virtual* llamada "Expression". Esta clase debe tener el metodo virtual `double evaluar()`.

### Tarea #2 (20%)

Declare la clase "Valor" la cual debe heredar de la clase "Expression". Esta clase debe:

- Tener un constructor que acepta un `double` llamado "valor"
- Su metodo "evaluar" debe retornar el "valor" que se le dio a su constructor

Esta clase pretende representar las expresiones numericas.

### Tarea #3 (20%)

Declare una *clase virtual* llamada "OperacionBinaria". Esta clase debe:

- Tener un constructor que acepta dos instancias de *Expression\** llamadas "operador1" y "operador2"
- Tiene un metodo virtual `double operar(const double op1, const double op2) const`
- Su metodo "evaluar" debe:
  1. Llamar al metodo "evaluar" de "operador1" y "operador2"
  2. Llamar a su metodo "operar" con los resultados del paso anterior
  3. Retornar el resultado de llamar al metodo "operar"

## Tarea #4 (20%)

Declare las clases “Suma” y “Multiplicacion”. Estas clases deben heredar de la clase “OperacionBinaria”. El metodo “operar” de la clase “Suma” debe sumar sus parametros y el metodo “operar” de la clase “Multiplicacion” debe multiplicar sus parametros.

## Tarea #5 (20%)

Defina una función “`bool parse(const std::string expression, Expression * resultado)`” la cual debe leer el string que se le dio como parametro y construir una instancia de la clase “Expresion” a partir de ese string. Por ejemplo:

```
int main() {
    std::string entrada("1+2*3");
    Expression* expr;

    // Si se llama a la funcion "parse" de la siguiente manera
    parse(entrada, expr);

    // Es equivalente a construir una expresoin de la siguiente manera
    Expression* expr2 = new Suma(new Valor(1), new Multiplicacion(new Valor(2), new Valor(3)));

    // Por lo tanto la siguiente linea imprime:
    // "La diferencia es 0.0"
    printf("La diferencia es %d", expr2->evaluar() - expr->evaluar());
}
```